Single-Image SVBRDF Estimation with Learned Gradient Descent

Luo, X.; Scandolo, L.; Bousseau, A.; Eisemann, E.

DOI
10.1111/cgf.15018

Publication date
2024

Document Version
Final published version

Published in
Computer Graphics Forum

Citation (APA)
Luo, X., Scandolo, L., Bousseau, A., & Eisemann, E. (2024). Single-Image SVBRDF Estimation with Learned Gradient Descent. *Computer Graphics Forum*, *43*(2), Article e15018. https://doi.org/10.1111/cgf.15018

DOI: 10.1111/cgf.15018
EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Single-Image SVBRDF Estimation with Learned Gradient Descent

X. Luo[1], L. Scandolo[1], A. Bousseau[1,2] and E. Eisemann[1]

[1]Delft University of Technology, The Netherlands
[2]Inria, Université Côte d'Azur, France

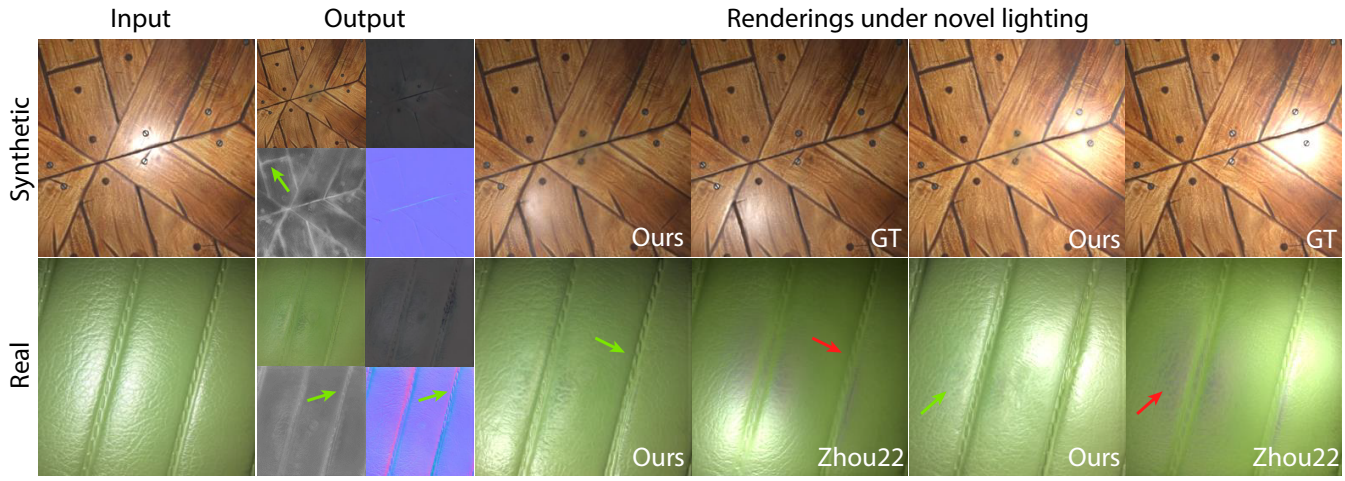

Figure 1: *Given a single flash picture as input, our learned gradient descent algorithm solves for SVBRDF maps that closely reproduce the input, while generalizing well to new view and light configurations. We demonstrate the effectiveness of our approach by comparing its output to ground-truth synthetic data (top) and to relit real images produced by state-of-the-art methods [ZK22] (bottom). Note how our method recovers intricate geometric details in the normal map, inpaints small saturated highlights with plausible material properties, and propagates roughness information away from the highlight.*

Abstract

Recovering spatially-varying materials from a single photograph of a surface is inherently ill-posed, making the direct application of a gradient descent on the reflectance parameters prone to poor minima. Recent methods leverage deep learning either by directly regressing reflectance parameters using feed-forward neural networks or by learning a latent space of SVBRDFs using encoder-decoder or generative adversarial networks followed by a gradient-based optimization in latent space. The former is fast but does not account for the likelihood *of the prediction,* i.e.*, how well the resulting reflectance explains the input image. The latter provides a strong* prior *on the space of spatially-varying materials, but this prior can hinder the reconstruction of images that are too different from the training data. Our method combines the strengths of both approaches. We optimize reflectance parameters to best reconstruct the input image using a recurrent neural network, which iteratively predicts how to update the reflectance parameters given the gradient of the reconstruction likelihood. By combining a learned prior with a likelihood measure, our approach provides a* maximum a posteriori *estimate of the SVBRDF. Our evaluation shows that this* learned gradient-descent *method achieves state-of-the-art performance for SVBRDF estimation on synthetic and real images.*

CCS Concepts

• ***Rendering*** → *Reflectance/Shading Models;*

1. Introduction

Real-world objects have a rich visual appearance due to spatially-varying material properties, which can be represented by

Spatially-Varying Bidirectional Reflectance Distribution Functions (SVBRDFs). This paper presents a lightweight method to capture the appearance of real surfaces with only a single photo.

Since few measurements are insufficient to ensure a unique interpretation of the many reflectance parameters, recent research leveraged deep learning to automatically build priors based on the distribution of plausible SVBRDFs. A first family of methods trains feed-forward neural networks to predict spatially-varying reflectance parameters from as little as a single flash picture of a flat surface [LDPT17; YLD*18; LSC18; DAD*18; VPS21; ZK21; GLT*21]. While fast, such neural networks mostly rely on data-driven priors to make their prediction. During use, they never evaluate the actual quality of their output with respect to the input image.

A second family of methods achieves higher accuracy by using differentiable rendering for online optimization, where the estimated SVBRDF is rendered under the same viewing and lighting conditions as for capture, Gradient descent is used to minimize the difference between the rendering and the input image. Yet, relying solely on differentiable rendering to optimize reflectance parameters is prone to bad minima, which is why several groups of authors proposed to perform the optimization in a lower-dimensional latent space learned from a dataset of representative SVBRDFs [GLD*19; GSH*20]. Nevertheless, gradient descent typically requires many iterations to converge and latent-space regularization can limit the quality of the estimation when the input differs too much from the images used to build the latent space.

Recent work proposed to combine these two strategies by training a neural network on a large dataset of SVBRDFs, and then fine-tuning the network weights at test time such that its prediction best reproduces the input image [FR22; ZK22]. A key challenge with this new strategy is to prevent the fine-tuning phase to forget the priors learned during the training phase.

Our algorithm combines the speed of neural network prediction with the accuracy of test-time optimization. It is inspired by *learned gradient descent* [ADG*16; PW17], which replaces the analytic gradient update rule of standard optimization by a recurrent neural network. This network is trained to predict the best updates given the current state of the estimation and the gradient of the cost function to be minimized. Importantly, the neural network weights are not updated at test-time, avoiding the risk of forgetting its priors. In our context, the cost function captures the *likelihood* of the SVBRDF to reproduce the input image when rendered under the same light and view, while the neural network learns a *prior* over the distribution of SVBRDFs. By combining likelihood and prior information, our method effectively solves for a *maximum a posteriori* estimate of the SVBRDF. While trained on a synthetic dataset of SVBRDFs, our method generalizes well to real data, outperforming both feed-forward and optimization-based prior work, as demonstrated on a large set of photographs.

2. Background and Related Work

We focus our discussion on recent deep learning methods for lightweight SVBRDF capture, and refer to surveys for a comprehensive overview of the vast domain of appearance acquisition [WLL*09; GGG*16; Don19]. We first introduce general concepts on which our approach relies, before diving into recent methods, which combine deep-learning and gradient-based optimization to recover SVBRDF parameters from one or a few flash images of a planar surface.

Appearance capture as an inverse problem. Formally, the image $\mathbf{I}$ of a surface depends on its reflectance properties $\mathbf{R}$, as well as the viewing conditions $\mathbf{V}$ and lighting conditions $\mathbf{L}$ under which the surface is captured:

$$\mathbf{I} = f(\mathbf{R}, \mathbf{L}, \mathbf{V}) + \mathbf{n}, \tag{1}$$

where f is the image formation model and $\mathbf{n}$ is measurement noise.

Appearance capture aims at inverting the image formation to recover $\mathbf{R}$ from observations $\mathbf{I}$, typically under known viewing and lighting conditions:

$$\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \mathcal{L}_{\text{reconstruct}}\left(\mathbf{I}, f(\mathbf{R}, \mathbf{L}, \mathbf{V})\right), \tag{2}$$

where $\mathcal{L}_{\text{reconstruct}}$ is a cost function measuring the difference between the observations and renderings of the estimated reflectance. Assuming that $\mathcal{L}_{\text{reconstruct}}$ is differentiable, gradient descent can be employed for the minimization:

$$\mathbf{R}_{t+1} = \mathbf{R}_t - \gamma_t \left.\frac{\partial \mathcal{L}_{\text{reconstruct}}\left(\mathbf{I}, f(\mathbf{R}, \mathbf{L}, \mathbf{V})\right)}{\partial \mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_t}, \tag{3}$$

where γ_t is the step size at iteration t.

To make this inverse problem well-posed, early work relied on dedicated gantries to capture many images of the target surface under different light and view configurations [MWL*99; Mat03; LKG*03; WGK14]. Despite progress in hardware setups and optimization algorithms [AWL13; KCW*18; ALL20], precise acquisition of spatially-varying materials remains a costly and time-consuming process. Moreover, gradient-based optimization often requires a large number of iterations and is subject to bad local minima, especially using few measurements.

Lightweight capture methods trade accuracy for simplicity to enable SVBRDF capture with as few as a single photograph of a surface – typically planar. Such methods compensate for the measurement scarcity by making various assumptions on the materials to be acquired, such as the existence of a low-dimensional basis of BRDFs [DWT*10; RWS*11; HSL*17; ZCD*16], or the presence of repetitive or stochastic patterns [AWL*15; WSM11].

Feed-forward SVBRDF prediction. Recent work shifted from hand-crafted assumptions towards priors learned from large datasets of (synthetic) SVBRDFs. A first family of methods cast SVBRDF acquisition as a regression task, for which they train a feed-forward neural network g_ω to directly predict reflectance properties from an input image [DAD*18; LDPT17; YLD*18; LSC18; GLT*21; ZK21]. Denoting $\{\tilde{\mathbf{R}}, \tilde{\mathbf{I}}\}$ a large set of SVBRDFs and their renderings, training the neural network with supervised learning amounts to solving for parameters ω, minimizing a loss function $\mathcal{L}_{\text{reflectance}}$, which compares the predicted SVBRDFs with the ground truth:

$$\hat{\omega} = \arg\min_{\omega} \sum_{\{\tilde{\mathbf{R}}, \tilde{\mathbf{I}}\}} \mathcal{L}_{\text{reflectance}}\left(g_\omega\left(\tilde{\mathbf{I}}\right), \tilde{\mathbf{R}}\right). \tag{4}$$

Further developments of such methods include the use of a rendering loss $\mathcal{L}_{\text{rendering}}\left(f\left(g_\omega\left(\tilde{\mathbf{I}}\right),\{\mathbf{L},\mathbf{V}\}\right),f\left(\tilde{\mathbf{R}},\{\mathbf{L},\mathbf{V}\}\right)\right)$ to evaluate whether the predicted SVBRDF has the same appearance as the ground truth under varying viewing and lighting conditions [DAD*18], or an adversarial loss $\mathcal{L}_{\text{adv}}\left(g_\omega\left(\tilde{\mathbf{I}}\right),\{\tilde{\mathbf{R}}\}\right)$ to evaluate whether the predicted SVBRDF resembles the ones in the dataset [GLT*21], or $\mathcal{L}_{\text{adv}}\left(f\left(g_\omega\left(\tilde{\mathbf{I}}\right),\mathbf{L},\mathbf{V}\right),\{\tilde{\mathbf{I}}\}\right)$ to evaluate whether the re-rendered image resembles synthetic and real images [VPS21; ZK21; ZWX*20].

Latent-space optimization. While feed-forward neural networks are fast to evaluate, the SVBRDF parameters they produce are entirely defined by the SVBRDF dataset $\{\tilde{\mathbf{R}}\}$ they are trained on, not by how well these parameters reproduce the input image $\mathbf{I}$ at test time. In other words, feed-forward networks only provide an approximate solution to the inverse problem formulated in Equation 2, and the severity of this approximation tends to increase for input images that deviate from the distribution of the training images $\{\tilde{\mathbf{I}}\}$. This discrepancy has motivated the development of test-time optimization methods that use gradient descent (Equation 3) to refine neural-network predictions to better fit the input images. Since SVBRDF recovery from few input images is ill-posed, several papers propose to regularize the problem by performing gradient descent in a low-dimensional SVBRDF latent space, instead of the original high-dimensional parameter space of $\mathbf{R}$ [GLD*19; GSH*20]:

$$\mathbf{z}_{t+1}=\mathbf{z}_t-\gamma_t\left.\frac{\partial\mathcal{L}_{\text{reconstruct}}\left(\mathbf{I},f\left(d_\psi\left(\mathbf{z}\right),\mathbf{L},\mathbf{V}\right)\right)}{\partial\mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_t}, \tag{5}$$

where a network d_ψ decodes the latent code $\mathbf{z}$ into an SVBRDF. Learning the latent space from a large dataset of SVBRDFs $\{\tilde{\mathbf{R}}\}$ ensures that the optimization produces plausible solutions. However, the optimization might struggle to find a latent code, which reproduces the input image well if it differs too much from the training data. Further, the many iterations required by gradient-based optimizations induce a significant overhead compared to direct prediction.

Network fine-tuning. Several authors proposed to fine-tune a feed-forward network g_ω at test time such that its prediction better reproduces the input [DDB20; ZK22; FR22], which amounts to performing gradient descent on the neural-network parameters rather than on the reflectance parameters or latent code:

$$\hat{\omega}=\arg\min_\omega\mathcal{L}_{\text{reconstruct}}\left(\mathbf{I},f\left(g_\omega\left(\mathbf{I}\right),\mathbf{L},\mathbf{V}\right)\right). \tag{6}$$

This strategy enables adjusting the prediction to the input, while still benefiting from the priors learned by the network during pre-training on a large dataset. Fischer and Ritschel [FR22] build on the concept of *meta-learning* to optimize the initialization of the network parameters and the gradient descent step sizes such that fine tuning converges quickly to good solutions. However, test-time fine-tuning runs the risk of forgetting the learned priors since it updates the weights by minimizing only the reconstruction error. A critical difference of our approach is to perform test-time optimization on the SVBRDF maps themselves, not on network weights, which ensures that the learned priors encoded by our recurrent neural network are preserved.

Similarly to meta-learning, Zhou and Kalantari [ZK22] propose to include fine-tuning steps during pre-training of the network, an algorithm they call *look-ahead training*. Yet, their approach also includes a secondary network that is trained to predict reflectance maps, which serves as a data-driven prior during test-time fine-tuning. Nevertheless, this prior is combined with the reconstruction error as a linear combination (Eq.7 in their paper) and is only used for the first iteration of the optimization (Sec.4.4 in their paper). In contrast, we provide the gradient of $\mathcal{L}_{\text{reconstruct}}$ to a recurrent network that learns to best combine this test-time information with its priors to iteratively improve the prediction.

Importantly, while [FR22] and [ZK22] rely on hand-tuned step sizes for the gradient descent optimization, our method predicts the magnitude of the steps and yields results of similar quality in much fewer steps, making it 10x faster than [ZK22] (Table 2).

3. Appearance capture with learned gradient descent

3.1. Problem formulation

Our approach combines the respective strengths of optimization-based and regression-based methods. We cast appearance capture as the minimization problem of Equation 2, using a single flash image $\mathbf{I}$ as observation of the planar surface to acquire. Yet, we replace the brittle and costly analytic gradient descent of Equation 3 by a *learned* gradient descent [ADG*16; PW17], where we train a recurrent neural network h_θ to predict how to progressively update an estimate $\mathbf{R}_t$ of the SVBRDF:

$$\mathbf{R}_{t+1}=\mathbf{R}_t-h_\theta\left(\left.\frac{\partial\mathcal{L}_{\text{reconstruct}}\left(\mathbf{I},f\left(\mathbf{R},\mathbf{L},\mathbf{V}\right)\right)}{\partial\mathbf{R}}\right|_{\mathbf{R}=\mathbf{R}_t},\mathbf{R}_t\right). \tag{7}$$

This formulation corresponds to a *maximum a posteriori estimation*, where the cost function $\mathcal{L}_{\text{reconstruct}}$ is proportional to the *likelihood* of the solution with respect to the input, while the neural network h_θ captures a *prior* on the distribution of SVBRDFs. Intuitively, the likelihood term encourages fidelity to the input, while the prior helps resolving ambiguities and prevents overfitting. This formulation has several advantages over existing work:

- In the absence of a prior, standard gradient descent (Equation 3) corresponds to *maximum likelihood estimation*, which is ill-posed when only a single input image is available. While network fine-tuning makes the problem better posed by initializing the optimization with a data-driven prediction, it runs the risk of forgetting the prior learned by the network if too many optimization steps are performed. In contrast, by combining the neural-network prior with test-time gradients of $\mathcal{L}_{\text{reconstruct}}$, our approach converges to a good solution in only a few steps. In addition, our approach does not require specifying a step size γ_t, as the magnitude of the update is implicitly predicted by h_θ.
- In the absence of a test-time likelihood term, feed-forward networks rely mostly on priors learned from the training data distribution (Equation 6) to predict SVBRDFs in a single step. In contrast, our network performs the simpler task of progressively improving a running estimate of the SVBRDF given gradient information about its likelihood. In practice, this online optimization scheme allows us to produce much more accurate results than feed-forward methods.

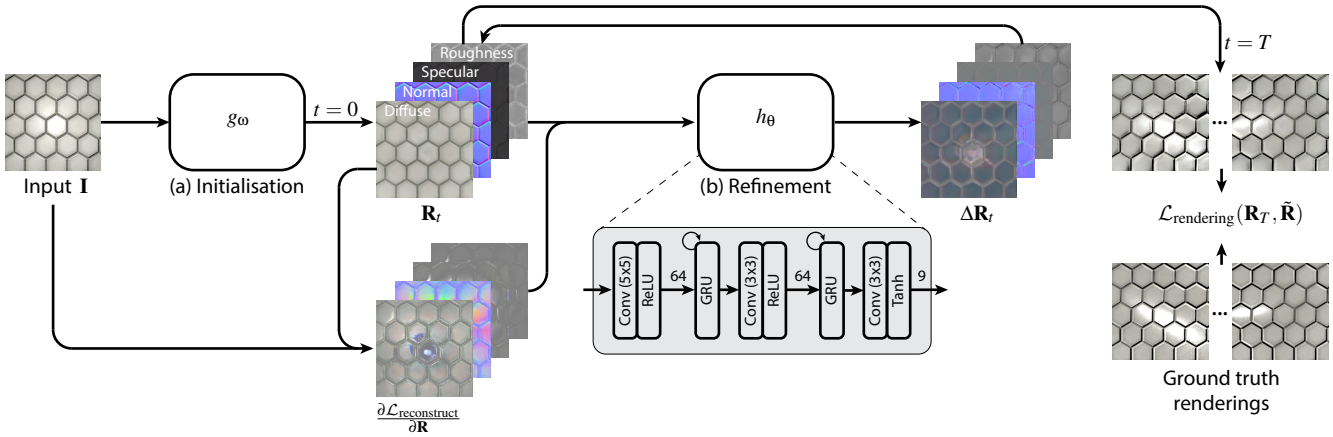

Figure 2: *Overview of our approach. The input image* $\mathbf{I}$ *is first fed to an existing feed-forward neural network* g_ω *to predict an initialization* $\mathbf{R}_0$ *of the SVBRDF maps (a). This prediction is then iteratively refined by our recurrent neural network* h_θ *(b). At each iteration, the current estimate* $\mathbf{R}_t$ *is compared to the input using a differentiable renderer. The gradient of this reconstruction loss,* $\frac{\partial \mathcal{L}_{reconstruct}}{\partial \mathbf{R}}$*, is fed to* h_θ *along with* $\mathbf{R}_t$*. The recurrent network predicts an update* $\Delta\mathbf{R}_t$ *of the SVBRDF, which is added to* $\mathbf{R}_t$ *to form the estimate* $\mathbf{R}_{t+1}$ *for the next iteration. Our algorithm performs* $T = 6$ *such iterations in practice. We train* g_ω *and* h_θ *jointly to minimize the difference between renderings of the final prediction* $\mathbf{R}_T$ *and renderings of the ground truth material maps under various view and light conditions. Note that the gradient and update images were scaled for visualization purpose.*

- While latent-space optimization methods benefit from data-driven priors, these priors are learned in a pre-process via auto-encoders [GLD*19] or generative-adversarial networks [GSH*20] trained on synthetic SVBRDFs. In contrast, our neural network learns priors by being trained specifically to perform maximum a posteriori estimation. As such, it accounts for the availability of the test-time likelihood. Importantly, our optimization happens in the original reflectance parameter space and is, thus, not limited to a pre-defined latent space.

3.2. Implementation

Our method belongs to the family of learned gradient-descent algorithms [ADG*16; PW17] that rely on recurrent neural networks to implement update rules that automatically leverage the inherent structure of the optimization problem at hand. While learned gradient descent has been successfully used to solve inverse imaging problems, such as novel-view synthesis [FBD*19] and MRI reconstruction [LPS*19; PW19], we make specific adaptations to apply this approach to single-image SVBRDF capture (see Fig. 2).

The core of our approach is a lightweight recurrent neural network h_θ that takes as input the current estimate of the SVBRDF $\mathbf{R}_t$ along with the gradient of the cost function $\mathcal{L}_{\text{reconstruct}}$ with respect to $\mathbf{R}_t$, which we obtain via automatic differentiation. The network outputs an update $\Delta\mathbf{R}_t$, which is summed with $\mathbf{R}_t$ to produce $\mathbf{R}_{t+1}$. In our implementation, we formulate $\mathcal{L}_{\text{reconstruct}}$ as the image difference between the input $\mathbf{I}$ and a rendering of $\mathbf{R}_t$ under a view and light setup that corresponds to a flash picture taken perpendicularly to the surface at a fixed distance. We use the L_2 norm to compute this difference, which corresponds to the log-likelihood under a Gaussian distribution assumption.

We initialize the SVBRDF estimate $\mathbf{R}_0$ by processing the input image $\mathbf{I}$ with the feed-forward network g_ω of Deschaintre et al. [DAD*18]. While we experimented with the pre-trained weights provided by the authors, we achieved better results by re-training this initialization network jointly with our recurrent updating network. We hypothesize that joint training enables the initialization network to account for the subsequent optimization performed by the recurrent network, similarly to the *meta-learning* and *look-ahead* strategies recently proposed by [FR22] and [ZK22] in the context of test-time network fine tuning.

Internally, h_θ is composed of three convolutional layers interleaved with Gated Recurrent Units (GRUs) [CVG*14]. The first two convolutional layers are activated with leaky ReLU functions and output feature maps of 64 channels, while the third convolutional layer is activated with a hyperbolic tangent to produce values between -1 and 1, which represent the update of the 9 SVBRDF channels, where 3 channels correspond to the diffuse albedo, 3 channels to the specular albedo, 2 channels to the normal, and 1 channel to the specular roughness. We used convolutional kernels of size 5×5 for the first layer and 3×3 for the second and third layer, resulting in $405,376$ parameters in total for h_θ, much less than the $159,741,922$ parameters of the initialization network g_ω. We voluntarily built on the classical UNet of Deschaintre et al. and on a lightweight recurrent network to demonstrate that the boost in performance achieved by our approach is due to methodological rather than architectural novelty.

An important hyper-parameter of our method is the number of iterations (or updates) T performed by the recurrent network. While several iterations are necessary to improve the prediction, performing too many iterations can be expensive in terms of GPU memory and time. Specifically, the GPU memory consumption of the recurrent network increases linearly with the number of time steps. Every iteration adds a forward pass through the CNN layers and the

computation of the gradient of the reconstruction error. Therefore, the training/testing time of the recurrent network also increases linearly with the number of time steps. We empirically found that $T = 6$ iterations offer a good trade-of, as detailed in Section 4.1.

3.3. Data and training

Similarly to prior work [DAD*18; GLD*19; GSH*20; GLT*21; ZK21; ZK22], we adopt a Cook-Torrance SVBRDF model [CT82] with the GGX distribution [WMLT07], which is parameterized by four material maps, corresponding to the diffuse/specular albedo, specular roughness, and surface normal. We visualize all inputs and results in gamma space, except normals and roughness, which we keep in linear space.

We train the initialization network g_ω and our recurrent update network h_θ jointly on the dataset of [DAD*18], which contains 99,533 synthetic SVBRDFs $\{\tilde{\mathbf{R}}\}$. We render the images $\{\tilde{\mathbf{I}}\}$ of these SVBRDFs under view $\mathbf{V}$ and light $\mathbf{L}$ that emulate a camera positioned perpendicularly and at a fixed distance to the planar surface, with a co-located flash of fixed intensity. We adjusted these parameters by hand to best reproduce the appearance of the renderings provided by [DAD*18]. We assume that the test-time input images are captured under similar view and light conditions, and thus use the same parameters to compute the gradient of $\mathcal{L}_{\text{reconstruct}}$ fed to h_θ. We train our method to minimize the rendering loss proposed by [DAD*18], which compares renderings of the material maps $\mathbf{R}_T$ predicted at the last iteration of our recurrent network with renderings of the ground-truth maps $\tilde{\mathbf{R}}$, under 9 random lighting and viewing conditions $\{\mathbf{L}, \mathbf{V}\}$. Following [DAD*18], we use the $L1$ norm and compare the logarithmic values of the renderings:

$$\mathcal{L}_{\text{rendering}}(\mathbf{R}_T, \tilde{\mathbf{R}}) = \sum_{\{\mathbf{L},\mathbf{V}\}} \left| \log f(\mathbf{R}_T, \mathbf{L}, \mathbf{V}) - \log f(\tilde{\mathbf{R}}, \mathbf{L}, \mathbf{V}) \right|. \tag{8}$$

We used the Adam optimizer with a learning rate set to 0.00002, betas set to (0.9, 0.999), and the weight decay set to 0. We trained our method until convergence (80 epochs with a batch size of 4), which took three weeks on an NVIDIA A40 GPU. Once trained, our method infers SVBRDF maps from an image in around 0.1 seconds on the same NVIDIA A40 GPU.

4. Ablation studies

We conducted several ablation studies to assess the impact of the number of iterations performed by our recurrent network, as well as the benefit of providing gradient information to this network at test time. Similarly to [ZK22], we performed all studies on a set of 61 synthetic SVBRDF, 22 being provided by [DAD*19] and 39 by [GSH*20]. Importantly, none of these SVBRDFs were used to generate the training data. We created synthetic flash inputs for this test set by rendering each SVBRDF under the same light and view conditions as the ones used for training our method.

We evaluate the quality of the prediction by comparing renderings of the SVBRDFs to ground truth in terms of root mean squared error (RMSE) and learned perceptual image patch similarity (LPIPS) [ZIE*18], averaged over 20 random light and view configurations that differ from the colocated flash configuration used to render the input.

4.1. Number of iterations

We first evaluate the performance of our recurrent network related to the number of iterations T. We trained different models with $T = 2$ to $T = 10$. Fig. 3(top left) plots the RMSE and LPIPS achieved by these models on the test set. This experiment reveals that while the RMSE saturates after 6 iterations, LPIPS increases slightly when more iterations are performed, even though it remains lower than the LPIPS achieved by previous methods (see Table 2). We thus fix the total number of iterations to 6, which offers a good trade-off between accuracy and complexity of the model. Fig. 3(top right) plots the evolution of the RMSE and LPIPS of the test set over the iterations of the model trained for 6 iterations, showing that quality improves as the optimization progresses. In practice, the magnitude of improvement varies between materials. Fig. 4 shows two typical SVBRDFs where the initial prediction is either too shiny, or not enough, and gets corrected by subsequent iterations. Finally, Fig. 3(bottom) plots the evolution of the same metric when we let the model trained on 6 iterations run for more iterations. While the error remains stable for up to 24 iterations, it does not decrease significantly, and it eventually increases if too many iterations are performed.

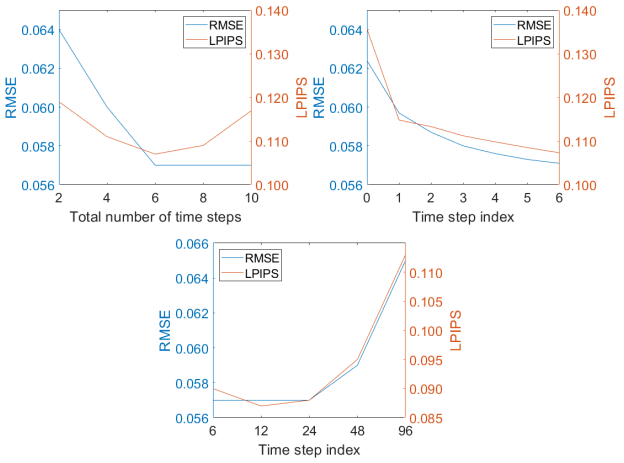

Figure 3: *Impact of the number of iterations performed by the recurrent network. Upper left: comparison between models trained with an increasing total number of iterations. Upper right: evolution of the accuracy achieved by a model trained for a total of 6 iterations. Lower middle: evolution of the accuracy in further inference steps with the same model trained for a total of 6 iterations*

.

4.2. Test-time gradient information

Our architecture improves upon the one proposed by [DAD*18] by complementing it with a recurrent network, and by providing test-time gradient information to that recurrent network. We now

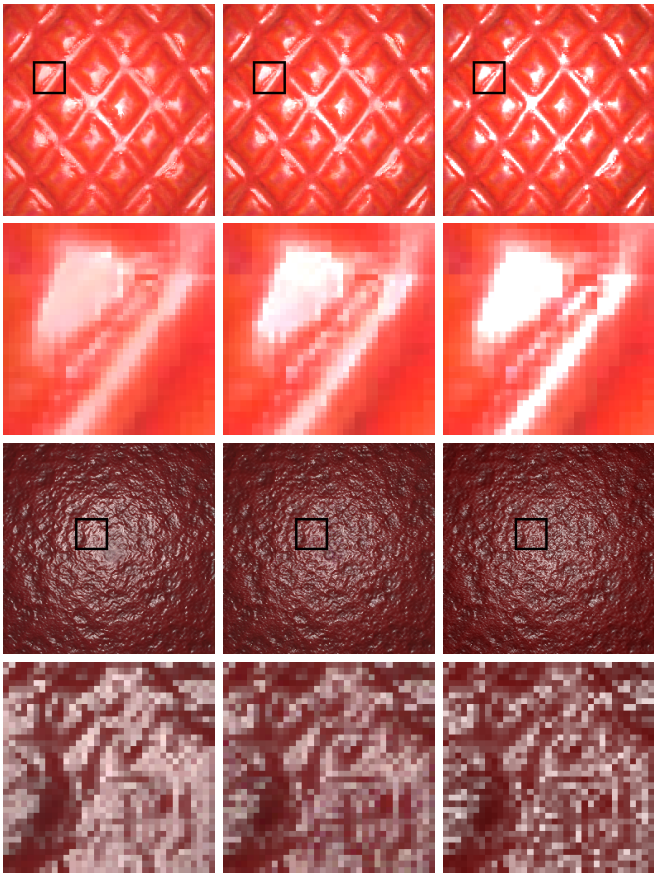

Figure 4: *Starting with the initial prediction (left), the recurrent network refines the result (middle), bringing it closer to GT (right). In these examples, the refinement mostly affects the intensity, spread and sharpness of the highlights to make the material more (top rows) or less (bottom rows) shiny.*

evaluate the impact of these two additional components. To do so, we compare the pre-trained model g_ω by [DAD*18] to two versions of our architecture.

The first version augments g_ω with the recurrent network h_θ, but only feeds this network with the intermediate prediction $\mathbf{R}_t$ at each iteration. The second and complete version feeds the recurrent network with $\mathbf{R}_t$ and the gradient $\frac{\partial \mathcal{L}_{\text{reconstruct}}}{\partial \mathbf{R}}$.

Table 1 summarizes the experiment's outcome. Complementing the architecture of [DAD*18] with a recurrent network already yields a significant increase in accuracy, which we attribute to the additional capacity that each iteration provides. Providing test-time gradient information to this recurrent network improves accuracy further, reducing RMSE by 31% and LPIPS by 52% over the baseline g_ω.

Table 1: *Ablation study to compare our complete method to the baseline architecture by [DAD*18], which does not include the recurrent network h_θ, and to a version that includes the recurrent network but no test-time gradient. RMSE and LPIPS of re-renderings are averaged over 20 random light/view configurations.*

	RMSE	LPIPS
without h_θ	0.083	0.223
without gradient	0.069	0.119
Ours	**0.057**	**0.107**

5. Results

We compare our approach to recent methods for lightweight SVBRDF capture, either based on feed-forward networks [DAD*18; ZK21; GLT*21] or on test-time optimization [GLD*19; GSH*20; ZK22]. We used the code and pre-trained weights provided by the authors of each method, except for [GLT*21] for which we sent our testing data to the authors, who kindly agreed to run their method and send back their results. We ran all methods on a single input image, even for methods that can process multiple images. We provide additional results, including animations under moving lights, as supplemental materials.

5.1. Comparison on synthetic images

We first focus on the synthetic test set (see Section 4). For all methods, we report the RMSE on the individual SVBRDF maps, as well as the RMSE and LPIPS errors on re-renderings averaged over 20 random light and view configurations. We use the same 20 configurations to compare all methods on a given SVBRDF. We generate these configurations by sampling the light and view positions uniformly over a quad of the same size as the surface patch, parallel to and above the surface. This ensures that the images always contain a highlight.

Table 2 summarizes the results achieved by each method[†]. When looking at individual maps, our method achieves the best result for diffuse albedo and normals, the second best result for specular albedo (outperformed by [ZK21]), and the third best result for roughness (outperformed by [GLT*21] and [ZK21]). Importantly, our method achieves the best results on re-renderings, both in terms of RMSE and LPIPS. Note also that our method is an order of magnitude slower than feedforward approaches [DAD*18;

[†] The numbers we report were computed by running all methods on our test set, which is composed of 61 synthetic materials provided by [DAD*19] and [GSH*20]. The difference between these numbers and the ones reported by Zhou and Kalantari [ZK22] might be due to the fact that their test set (which is not available) only contains 52 of our 61 materials, and that the viewing and lighting conditions we used to render the dataset might differ from the ones used by [ZK22] (which are unknown to us). Also, we observed that the synthetic inputs and roughness maps provided in [ZK22] are visually different from ours, which suggests that they treated the roughness maps from [GSH*20] as linear while we treated them as gamma-corrected to agree with the ones from [DAD*19]. Nevertheless, the RMSE and LPIPS values reported in Table 1 of [ZK22] remain suboptimal to ours on re-renderings.

Table 2: *Quantitative comparison on synthetic SVBRDFs. Des18, Guo21 and Zhou21 are fast feed-forward methods, while Gao19, Guo20 and Zhou22 are slower due to test-time optimization. Our approach achieves state-of-the-art quality while being an order of magnitude faster than the fastest optimization method. All timings were measured on an NVIDIA GeForce GTX 1080Ti GPU.*

	RMSE										LPIPS		Speed/sec
	Diffuse		Specular		Rough		Normal		Render		Render		
Method	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	
Des18	0.056	0.066	0.144	0.106	0.350	0.313	0.064	**0.032**	0.083	0.048	0.223	0.129	0.07
Guo21	0.094	0.115	0.119	0.128	**0.185**	**0.158**	0.071	0.039	0.095	0.054	0.214	0.090	NA
Zhou21	0.086	0.039	**0.089**	**0.077**	0.193	0.196	0.067	0.034	0.112	0.039	0.150	0.073	0.02
Gao19	0.070	0.040	0.119	0.087	0.296	0.279	0.073	0.035	0.084	0.035	0.139	0.076	42.70
Guo20	0.064	0.042	0.101	0.090	0.325	0.275	0.077	0.041	0.072	0.034	0.167	0.078	261.50
Zhou22	0.081	0.086	0.142	0.115	0.209	0.170	0.066	0.034	0.094	0.049	0.186	0.102	4.30
Ours	**0.051**	**0.035**	0.101	0.096	0.199	0.230	**0.061**	0.033	**0.057**	**0.032**	**0.107**	**0.070**	0.20

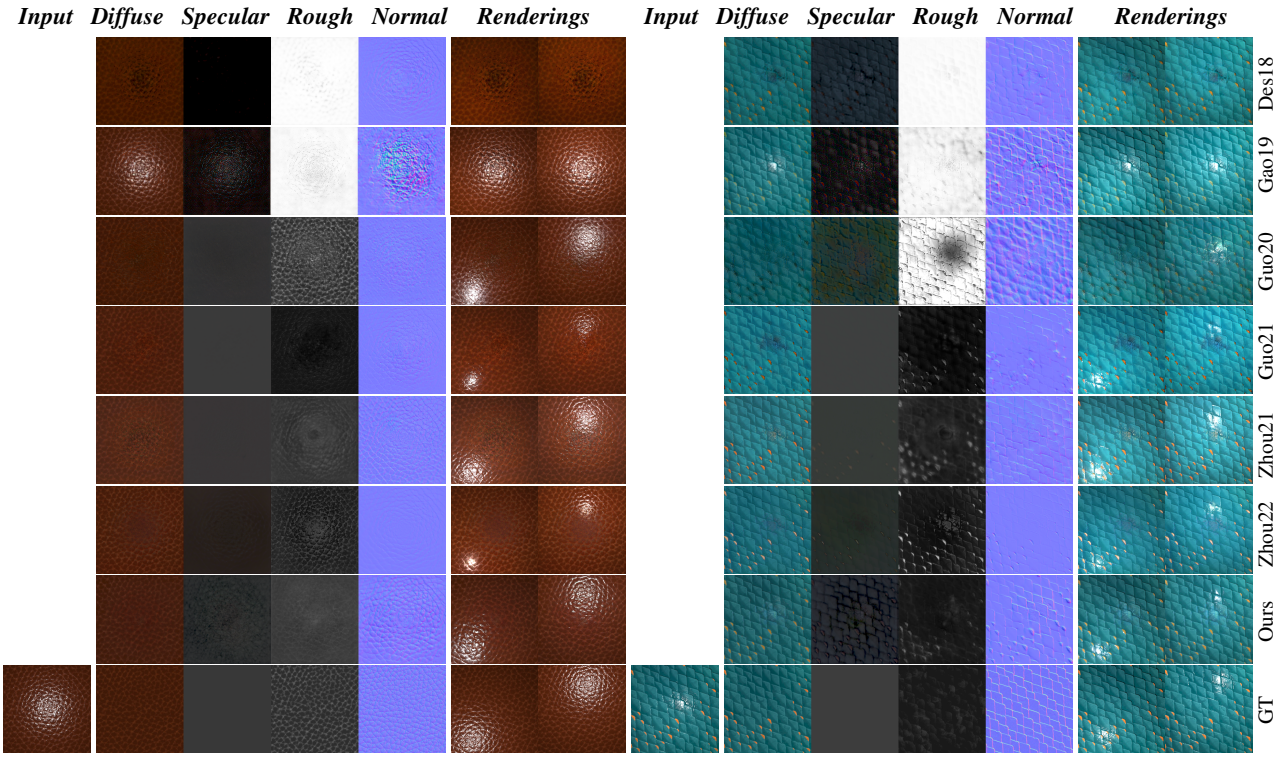

Figure 5: *Visual comparison against other methods on synthetic images. Note how our method recovers more faithful normal maps, as well as roughness and specular information away from the highlight. All images except roughness and normal maps are shown in gamma space for visualization.*

ZK21; GLT*21], but an order faster than fine-tuning [ZK22] and two to three orders faster than latent-space optimization [GLD*19; GSH*20].

Fig. 5 provides a visual comparison on two representative SVBRDFs. Overall, our approach based on learned gradient descent recovers finer details in the normal maps, including away from the highlight, and better reproduces the colors and contrast of the input.

5.2. Comparison on real images

We further examined a test set of 109 real scenes gathered by [ZK22], composed of 33 scenes by [GSH*20] and 76 by [ZK22]. Each scene has been captured under 9 calibrated view/light conditions, allowing us to use the central condition as input and the 8 other images as ground truth to compare re-renderings of the predicted SVBRDFs. To compute the test-time gradient for our method, we set the light intensity to be the same as during training and we assume that the camera, as well as the co-located light,

Table 3: *Quantitative comparison on real images, where one image serves as input and 8 other images are compared against re-renderings of the predicted SVBRDF.*

	Guo20		Zhou22	
Method	RMSE	LPIPS	RMSE	LPIPS
Des18	0.140	0.391	0.102	0.316
Gao19	0.158	0.361	0.110	0.290
Guo20	0.153	0.316	0.113	0.256
Guo21	0.161	0.391	0.103	0.303
Zhou21	0.154	0.314	0.132	0.266
Zhou22	0.133	0.286	0.093	0.216
Ours	**0.122**	**0.276**	**0.084**	**0.211**

are oriented perpendicularly the surface, even if this only approximately holds in practice.

Since the data and metrics are the same as the ones used by [ZK22] for their evaluation, we report their numbers in Table 3, along with our results, all of which were obtained by providing a single image as input to the different methods. Our method achieves the best results in terms of both RMSE and LPIPS, demonstrating its ability to generalize to real images despite being trained on synthetic data.

Fig. 8 provides a visual comparison to the most recent method by [ZK22] on six images including wood, ceramic, stone, canvas, and plaster. Our method is especially good at recovering details in the normal map, and at propagating roughness information away from the highlight. Comparisons on more images can be found in the supplementary materials.

We provide as supplemental materials a comparison with others on 93 flash photographs from [DAD*18], [GLT*21], [ZK22], as well as images we captured ourselves with a hand-held consumer-level camera. For a fair comparison, all optimization-based methods were executed with their default light and view parameters as input. The initialization for [GLD*19] was obtained by running [DAD*18]. Fig. 9 illustrates some of these results. We show a re-rendering of the SVBRDF under the same lighting conditions as the input, as well as a re-rendering under novel lighting. Compared to others, our approach better reproduces the input (details in the normal map, color and contrast, extent of the highlight) and generalizes well to novel light with little residual of the highlight in the individual maps.

6. Limitations, extensions and future work

While we observed that learned gradient descent helps inpainting saturated pixels (Fig. 1, Fig. 8 top row), the quality of the prediction degrades for large highlights, where a lot of information is lacking (Fig. 6, top). Similarly, while test-time optimization helps the method generalize beyond its training set, it is challenged by input images that are too far from the expected capture conditions. The bottom part of Fig. 6 illustrates such as case, where the input image is captured under a light source that is far from the expected collocated flash, yielding worse results than when collocated lighting is used.

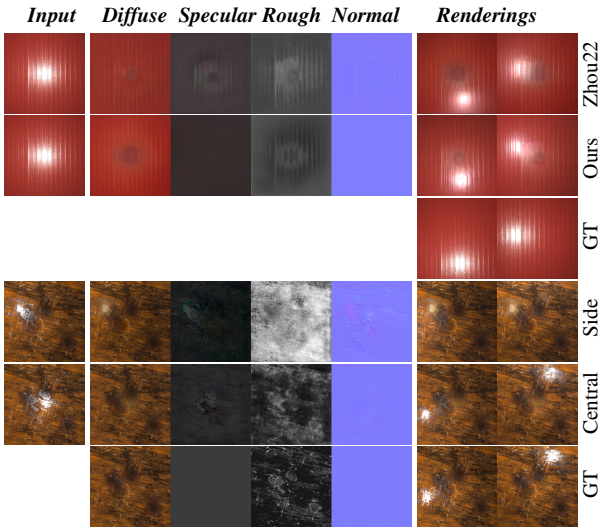

Figure 6: *Limitations. Top: Our method struggles to inpaint saturated pixels over large highlights; a limitation shared by existing single-image methods. Bottom: Our method assumes a collocated flash light, thus, prediction quality degrades when the material is captured under a side light (bottom).*

An exciting direction to address these limitations is to extend our optimization framework beyond single-image capture. Specifically, Equation 7 can be easily extended to compute $\mathcal{L}_{\text{reconstruct}}$ over multiple input images $\{\mathbf{I}\}$. As a first step in this direction, we adapted our method to take 5 images as input, taken under varying lighting and viewing conditions. Implementing this extension only requires modifying the initialisation network g_ω and the refinement network h_θ to process 5 images and 5 sets of gradient maps, respectively. While this extension increases the number of input channels of the network from 2×9 to $(N+1) \times 9$ for N inputs, we kept the subsequent dimensions fixed (64, 64 and 9 channels). We trained this extended architecture with the same synthetic data as in Section 3.3, except that we rendered each SVBRDF under 5 configurations of light and view positions, which we selected at random among 9 pre-defined configurations. We used the same test set of SVBRDFs as in Section 4 to compare this extension (Ours-multi) to our single-image model (Ours-single) and to the state-of-the-art multi-image optimization MaterialGAN [GSH*20] (Guo20-multi). Fig. 7 shows that our multi-image model outperforms [GSH*20] as well as our single-image model. In particular, having access to multiple images with different highlights helps recover material maps free of highlight residuals. Table 4 quantifies this improvement in terms of RMSE and LPIPS. Note that our multi-image model is only twice slower than the single-image model, while it is 600× faster than the latent-space optimization of MaterialGAN.

While these preliminary results are promising, handling real-world multi-image data would require training our method with more diverse light and view configurations. Moreover, robustness to approximate light and view calibration might be achieved by treating the per-image light and view parameters ($\mathbf{L}$,$\mathbf{V}$) as additional unknowns to be optimized along with the material maps $\mathbf{R}$.

Table 4: *Quantitative comparison of our multi-image extension (Ours-multi) against MaterialGAN [GSH*20] and our single-image model (Ours-single) on synthetic images.*

	RMSE		LPIPS		Speed/sec
Method	mean	std	mean	std	
Guo20-multi	0.068	0.030	0.148	0.071	261.50
Ours-single	0.057	0.032	0.107	0.070	0.20
Ours-multi	0.042	0.029	0.074	0.082	0.40

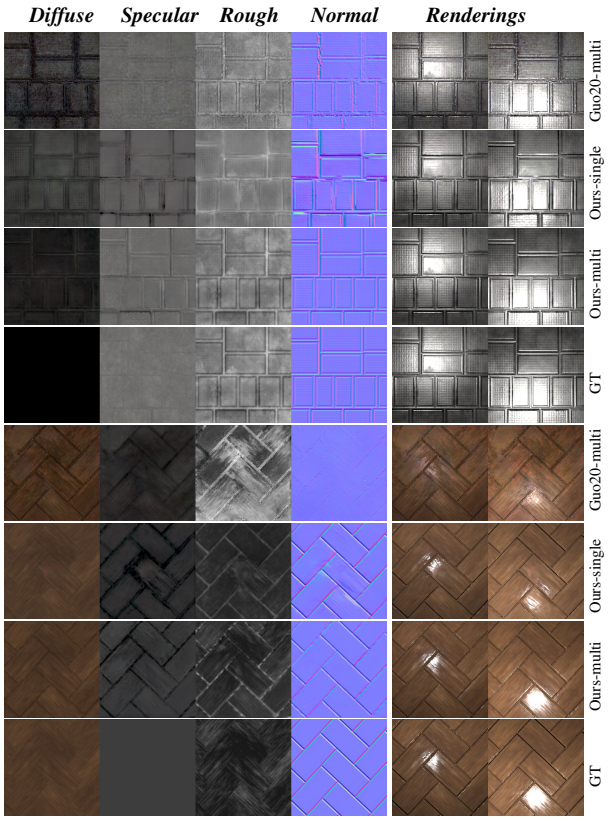

Figure 7: *Visual comparison between MaterialGAN [GSH*20], our single-image model and our multi-image model on synthetic images.*

7. Conclusion

Gradient descent is at the core of many inverse rendering algorithms, yet typically requires many steps and complementary regularization terms to converge to high-quality minima. We showed how *learned gradient descent* is well adapted to appearance capture, where the inherent structure of the problem can be leveraged by a neural network to perform gradient descent in a few high-quality steps. Intuitively, our recurrent neural network learns a prior about material appearance, while the forward rendering model gives a likelihood of reproducing the input. Feeding the network with the gradient of this rendering model effectively enables our method to solve for a maximum a posteriori estimate of the inverse problem of single-image SVBRDF capture. We also showed that the same formulation can be easily extended to a multi-image capture scenario. We strongly believe that a similar approach could benefit related inverse problems for which strong priors can be learned, such as facial and body capture, where feed-forward networks [KE18; SSSJ20] could be augmented with test-time optimization.

Acknowledgements

This work was partially funded by the NWO VIDI grant NextView, and was partially done while Adrien Bousseau was hosted by TU Delft for a year, supported by the Inria sabbatical exchange program. We would like to thank Valentin Deschaintre from Adobe Research, Jie Guo and Shuichang Lai from Nanjing University, and Yu Guo from University of California, Irvine for helping reproduce their results.

References

[ADG*16] Andrychowicz, Marcin, Denil, Misha, Gomez, Sergio, et al. "Learning to learn by gradient descent by gradient descent". *Advances in neural information processing systems* 29 (2016) 2–4.

[ALL20] Asselin, Louis-Philippe, Laurendeau, Denis, and Lalonde, Jean-François. "Deep SVBRDF estimation on real materials". *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, 1157–1166 2.

[AWL*15] Aittala, Miika, Weyrich, Tim, Lehtinen, Jaakko, et al. "Two-shot SVBRDF capture for stationary materials." *ACM Transactions on Graphics (Proc. SIGGRAPH)* 34.4 (2015), 110–1 2.

[AWL13] Aittala, Miika, Weyrich, Tim, and Lehtinen, Jaakko. "Practical SVBRDF capture in the frequency domain." *ACM Trans. Graph.* 32.4 (2013), 110–1 2.

[CT82] Cook, Robert L and Torrance, Kenneth E. "A reflectance model for computer graphics". *ACM Transactions on Graphics (ToG)* 1.1 (1982), 7–24 5.

[CVG*14] Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". *arXiv preprint arXiv:1406.1078* (2014) 4.

[DAD*18] Deschaintre, Valentin, Aittala, Miika, Durand, Fredo, et al. "Single-image svbrdf capture with a rendering-aware deep network". *ACM Transactions on Graphics (ToG)* 37.4 (2018), 1–15 2–6, 8.

[DAD*19] Deschaintre, Valentin, Aittala, Miika, Durand, Frédo, et al. "Flexible svbrdf capture with a multi-image deep network". *Computer Graphics Forum* 38.4 (2019), 1–13 5, 6.

[DDB20] Deschaintre, Valentin, Drettakis, George, and Bousseau, Adrien. "Guided Fine-Tuning for Large-Scale Material Transfer". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 39.4 (2020) 3.

[Don19] Dong, Yue. "Deep appearance modeling: A survey". *Visual Informatics* 3.2 (2019), 59–68 2.

[DWT*10] Dong, Yue, Wang, Jiaping, Tong, Xin, et al. "Manifold bootstrapping for SVBRDF capture". *ACM Transactions on Graphics (TOG)* 29.4 (2010), 1–10 2.

[FBD*19] Flynn, John, Broxton, Michael, Debevec, Paul, et al. "DeepView: View Synthesis With Learned Gradient Descent". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 4.

[FR22] Fischer, Michael and Ritschel, Tobias. "Metappearance: Meta-Learning for Visual Appearance Reproduction". *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 41.4 (2022) 2–4.

[GGG*16] GUARNERA, DARYA, GUARNERA, GIUSEPPE CLAUDIO, GHOSH, ABHIJEET, et al. "BRDF representation and acquisition". *Computer Graphics Forum* 35.2 (2016), 625–650 2.

[GLD*19] GAO, DUAN, LI, XIAO, DONG, YUE, et al. "Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images". *ACM Transactions on Graphics (Proc. SIGGRAPH)* 38.4 (2019), 134–1 2–8.

[GLT*21] GUO, JIE, LAI, SHUICHANG, TAO, CHENGZHI, et al. "Highlight-aware two-stream network for single-image SVBRDF acquisition". *ACM Transactions on Graphics (TOG)* 40.4 (2021), 1–14 2, 3, 5–8.

[GSH*20] GUO, YU, SMITH, CAMERON, HAŠAN, MILOŠ, et al. "MaterialGAN: Reflectance Capture Using a Generative SVBRDF Model". *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 39.6 (2020) 2–9.

[HSL*17] HUI, ZHUO, SUNKAVALLI, KALYAN, LEE, JOON-YOUNG, et al. "Reflectance capture using univariate sampling of brdfs". *Proceedings of the IEEE International Conference on Computer Vision*. 2017, 5362–5370 2.

[KCW*18] KANG, KAIZHANG, CHEN, ZIMIN, WANG, JIAPING, et al. "Efficient reflectance capture using an autoencoder." *ACM Trans. Graph.* 37.4 (2018), 127–1 2.

[KE18] KANAMORI, YOSHIHIRO and ENDO, YUKI. "Relighting Humans: Occlusion-Aware Inverse Rendering for Full-Body Human Images". *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 37.6 (2018) 9.

[LDPT17] LI, XIAO, DONG, YUE, PEERS, PIETER, and TONG, XIN. "Modeling surface appearance from a single photograph using self-augmented convolutional neural networks". *ACM Transactions on Graphics (Proc. SIGGRAPH)* 36.4 (2017), 1–11 2.

[LKG*03] LENSCH, HENDRIK PA, KAUTZ, JAN, GOESELE, MICHAEL, et al. "Image-based reconstruction of spatial appearance and geometric detail". *ACM Transactions on Graphics (TOG)* 22.2 (2003), 234–257 2.

[LPS*19] LØNNING, KAI, PUTZKY, PATRICK, SONKE, JAN-JAKOB, et al. "Recurrent inference machines for reconstructing heterogeneous MRI data". *Medical image analysis* 53 (2019), 64–78 4.

[LSC18] LI, ZHENGQIN, SUNKAVALLI, KALYAN, and CHANDRAKER, MANMOHAN. "Materials for masses: SVBRDF acquisition with a single mobile phone image". *Proceedings of the European conference on computer vision (ECCV)*. 2018, 72–87 2.

[Mat03] MATUSIK, WOJCIECH. "A data-driven reflectance model". PhD thesis. Massachusetts Institute of Technology, 2003 2.

[MWL*99] MARSCHNER, STEPHEN R, WESTIN, STEPHEN H, LAFORTUNE, ERIC PF, et al. "Image-based BRDF measurement including human skin". *Rendering Techniques' 99: Proceedings of the Eurographics Workshop in Granada, Spain, June 21–23, 1999 10*. Springer. 1999, 131–144 2.

[PW17] PUTZKY, PATRICK and WELLING, MAX. "Recurrent inference machines for solving inverse problems". *arXiv preprint arXiv:1706.04008* (2017) 2–4.

[PW19] PUTZKY, PATRICK and WELLING, MAX. "Invert to learn to invert". *Advances in neural information processing systems* 32 (2019) 4.

[RWS*11] REN, PEIRAN, WANG, JIAPING, SNYDER, JOHN, et al. "Pocket reflectometry". *ACM Transactions on Graphics (TOG)* 30.4 (2011), 1–10 2.

[SSSJ20] SAITO, SHUNSUKE, SIMON, TOMAS, SARAGIH, JASON, and JOO, HANBYUL. "PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization". *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 9.

[VPS21] VECCHIO, GIUSEPPE, PALAZZO, SIMONE, and SPAMPINATO, CONCETTO. "SurfaceNet: Adversarial SVBRDF Estimation from a Single Image". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 12840–12848 2, 3.

[WGK14] WEINMANN, MICHAEL, GALL, JUERGEN, and KLEIN, REINHARD. "Material classification based on training data synthesized using a BTF database". *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*. Springer. 2014, 156–171 2.

[WLL*09] WEYRICH, TIM, LAWRENCE, JASON, LENSCH, HENDRIK PA, et al. "Principles of appearance acquisition and representation". *Foundations and Trends® in Computer Graphics and Vision* 4.2 (2009), 75–191 2.

[WMLT07] WALTER, BRUCE, MARSCHNER, STEPHEN R, LI, HONGSONG, and TORRANCE, KENNETH E. "Microfacet models for refraction through rough surfaces". *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 2007, 195–206 5.

[WSM11] WANG, CHUN-PO, SNAVELY, NOAH, and MARSCHNER, STEVE. "Estimating Dual-Scale Properties of Glossy Surfaces from Step-Edge Lighting". *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 30.6 (2011) 2.

[YLD*18] YE, WENJIE, LI, XIAO, DONG, YUE, et al. "Single image surface appearance modeling with self-augmented cnns and inexact supervision". *Computer Graphics Forum* 37.7 (2018), 201–211 2.

[ZCD*16] ZHOU, ZHIMING, CHEN, GUOJUN, DONG, YUE, et al. "Sparse-as-possible SVBRDF acquisition". *ACM Transactions on Graphics (TOG)* 35.6 (2016), 1–12 2.

[ZIE*18] ZHANG, RICHARD, ISOLA, PHILLIP, EFROS, ALEXEI A, et al. "The unreasonable effectiveness of deep features as a perceptual metric". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 586–595 5.

[ZK21] ZHOU, XILONG and KALANTARI, NIMA KHADEMI. "Adversarial Single-Image SVBRDF Estimation with Hybrid Training". *Computer Graphics Forum* 40.2 (2021), 315–325 2, 3, 5–7.

[ZK22] ZHOU, XILONG and KALANTARI, NIMA KHADEMI. "Look-Ahead Training with Learned Reflectance Loss for Single-Image SVBRDF Estimation". *ACM Transactions on Graphics (TOG)* 41.6 (2022), 1–12 1–8, 11.

[ZWX*20] ZHAO, YEZI, WANG, BEIBEI, XU, YANNING, et al. "Joint SVBRDF Recovery and Synthesis From a Single Image using an Unsupervised Generative Adversarial Network." *EGSR (DL)*. 2020, 53–66 3.

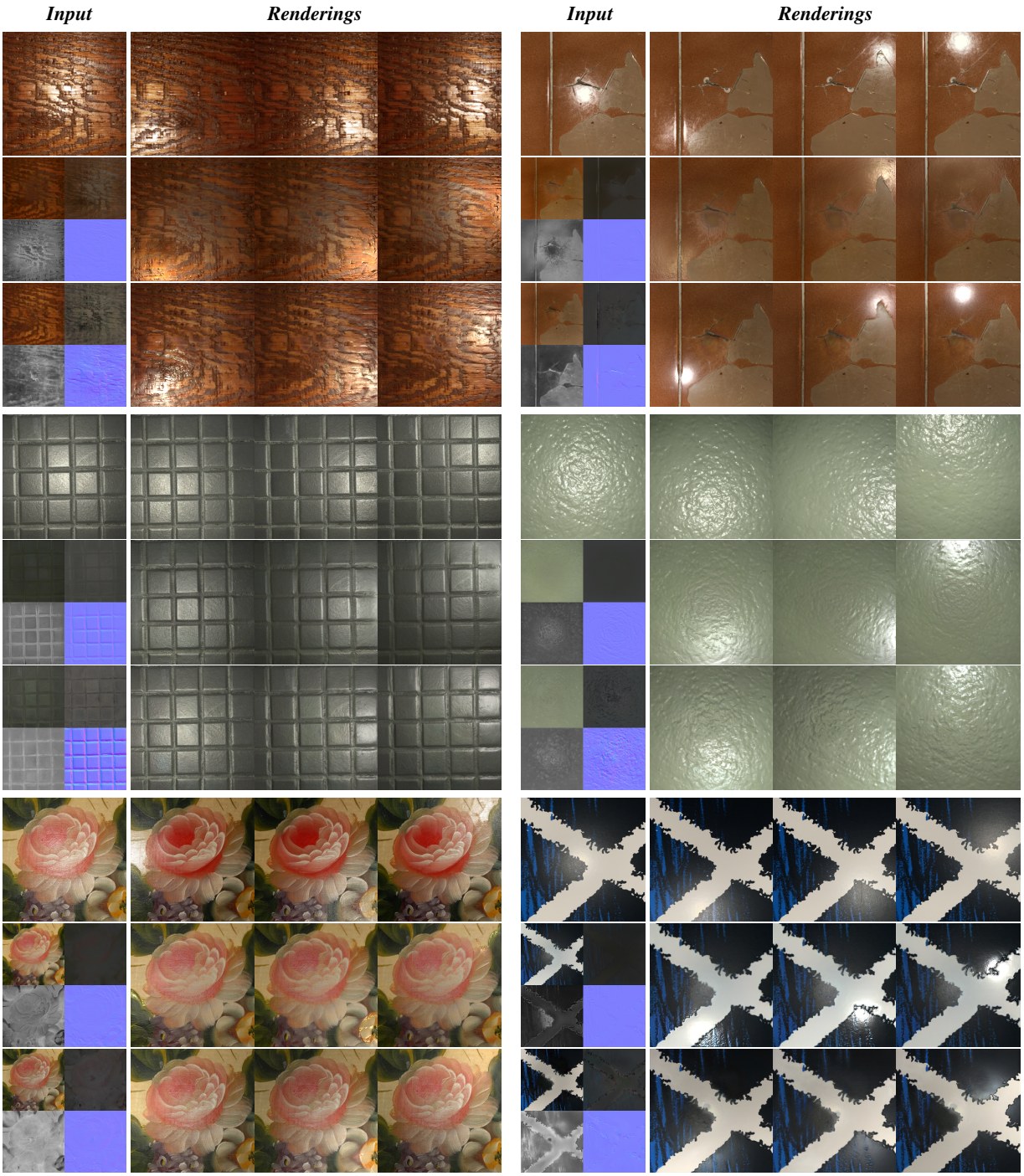

Figure 8: *Visual comparison with [ZK22] on real images with ground truth relighting. Note the fine geometric details in the normal maps and the propagation of spatially-varying roughness, which result in better reproduction of the ground truth appearance under novel lighting.*

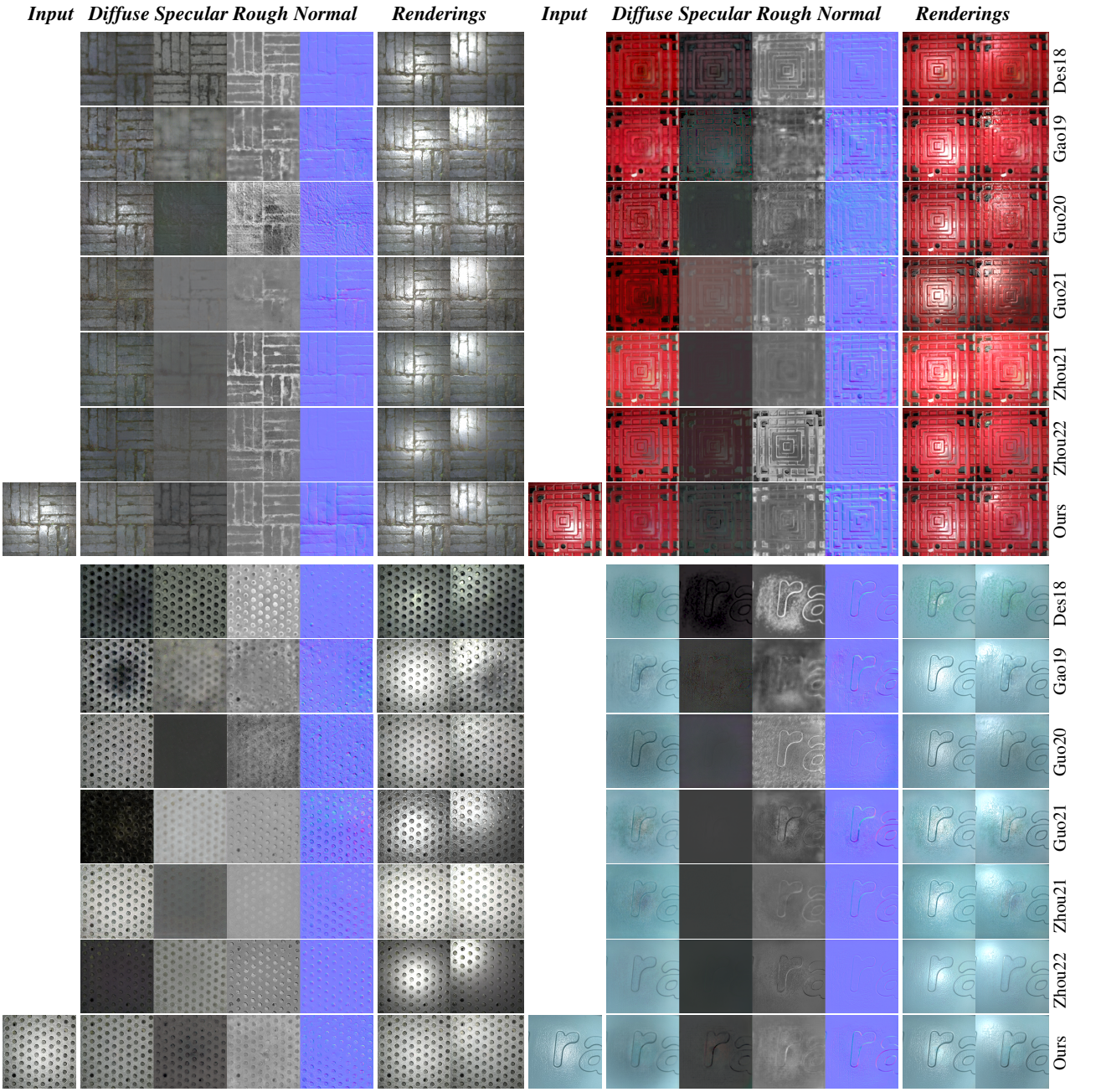

Figure 9: *Comparison with other methods on four real images. Our SVBRDFs reproduce well the input images when re-rendered under the same lighting conditions, and produce plausible novel relighting thanks to detailed normal maps and propagation of diffuse albedo and roughness within and away from the highlight respectively.*